



... for a brighter future



U.S. Department
of Energy



THE UNIVERSITY OF
CHICAGO



**Office of
Science**

U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory
managed by The University of Chicago

Scientific Software, Java, and Eclipse

Kenneth Evans, Jr.

Presented at the EPICS Collaboration Meeting

April 23 - 27, 2007

Deutsches Elektronen Synchrotron DESY, Hamburg, Germany

Outline

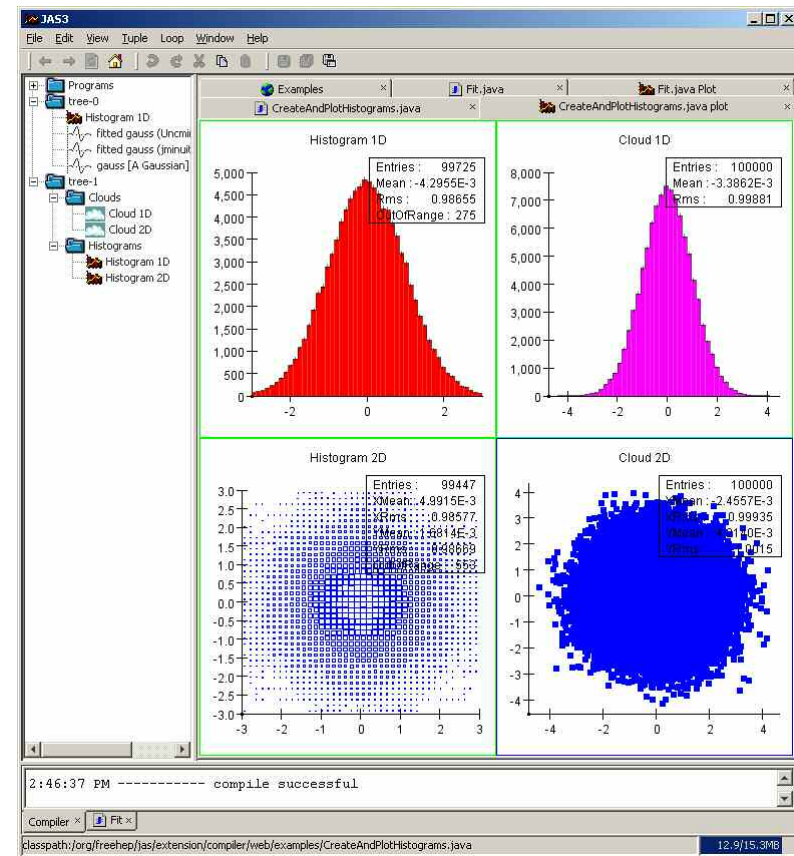
- Scientific Software and Examples
- Java
- X-Ray Software Development at the APS
- Eclipse and Examples

Scientific Software

- The language of choice used to be FORTRAN
 - There are still many legacy FORTRAN codes in use
- C and C++ have become popular
 - Grid computing now tends to be done in C
- Many scientists use Python
 - Reasonably powerful, yet easy to use
 - Allows them to do science rather than software
- There are now a number of significant scientific projects using Java
 - Many started out as C, but have evolved to Java
- Java is now an acceptable, if not the preferred, language for scientific software development

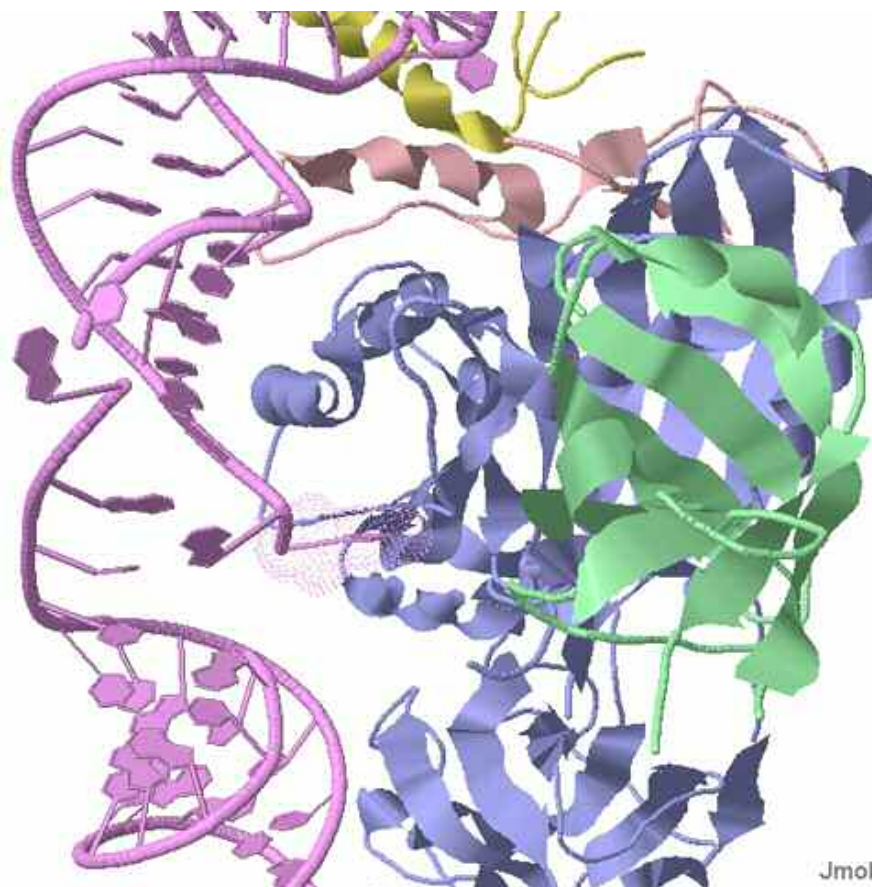
Java Analysis Studio (JAS3)

- Developed by and for the High-Energy physics community
- Plotting of 1d, 2d, 3d Histograms, XY plots, Scatter plots, etc.
- Open source
- Attractive plotting
- Fitting, other mathematical analysis
 - Primarily from CERN
- Highly modular structure
 - Uses plug-ins



JMol – Molecular Viewer

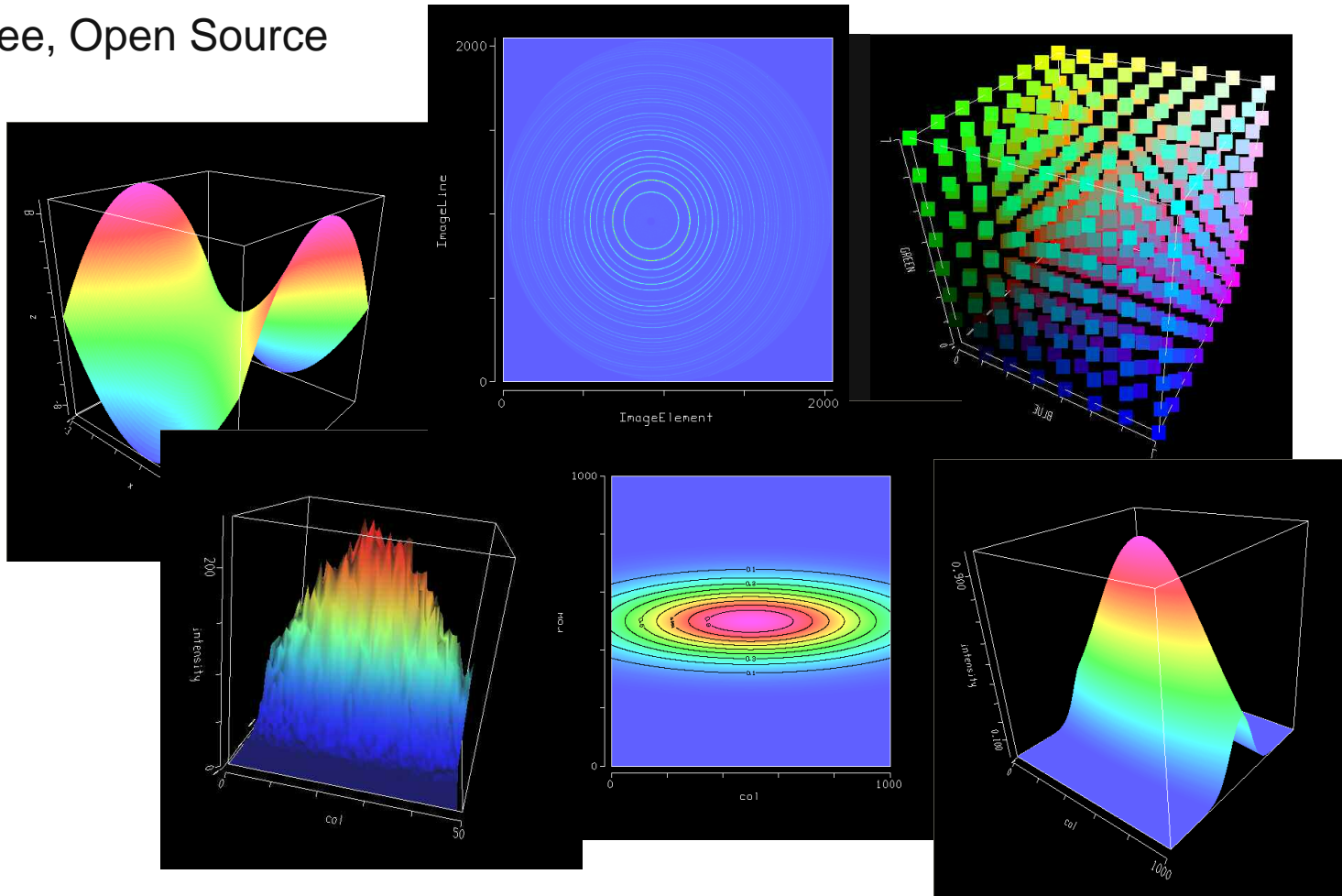
- Commonly used as an applet that can be integrated into web pages to display molecules in a variety of ways
- Also has a standalone application and a development tool kit that can be integrated into other Java applications
- Interactive, 3D
- Free, Open Source
- One of several Java Molecular Graphics packages



Crystal structure of an H/ACA box RNP from *Pyrococcus furiosus* (PDB CODE: 2HVY)

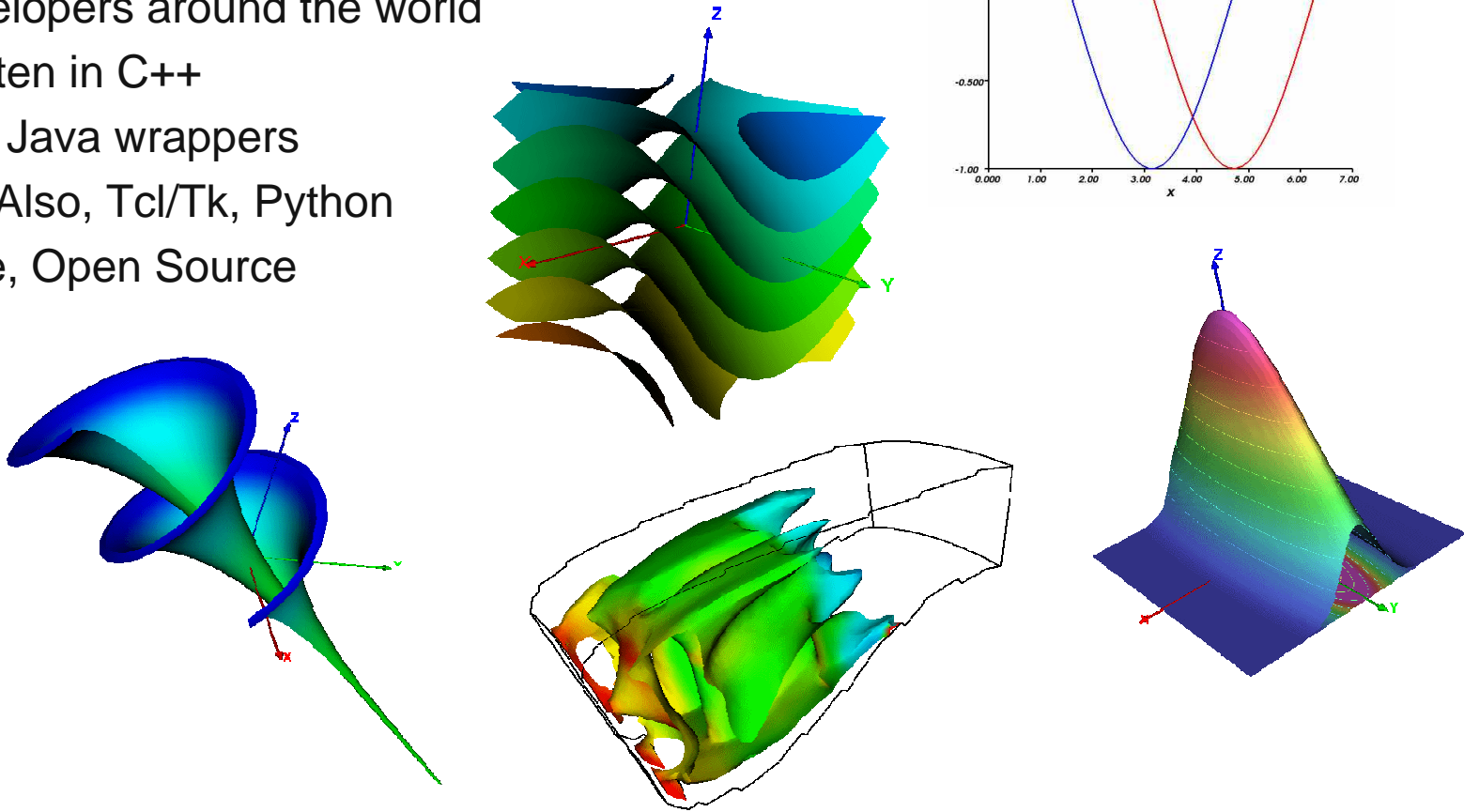
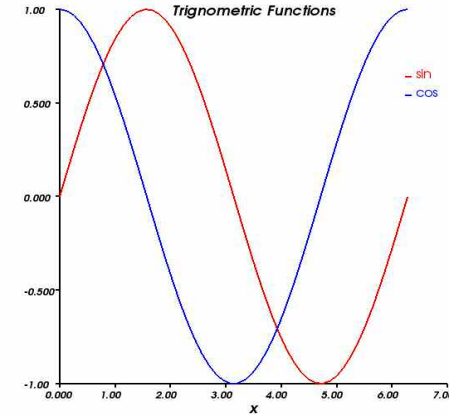
VisAD

- Space Sciences and Engineering Center (SSEC), and others
- Extensive 2D and 3D visualization package
- Free, Open Source



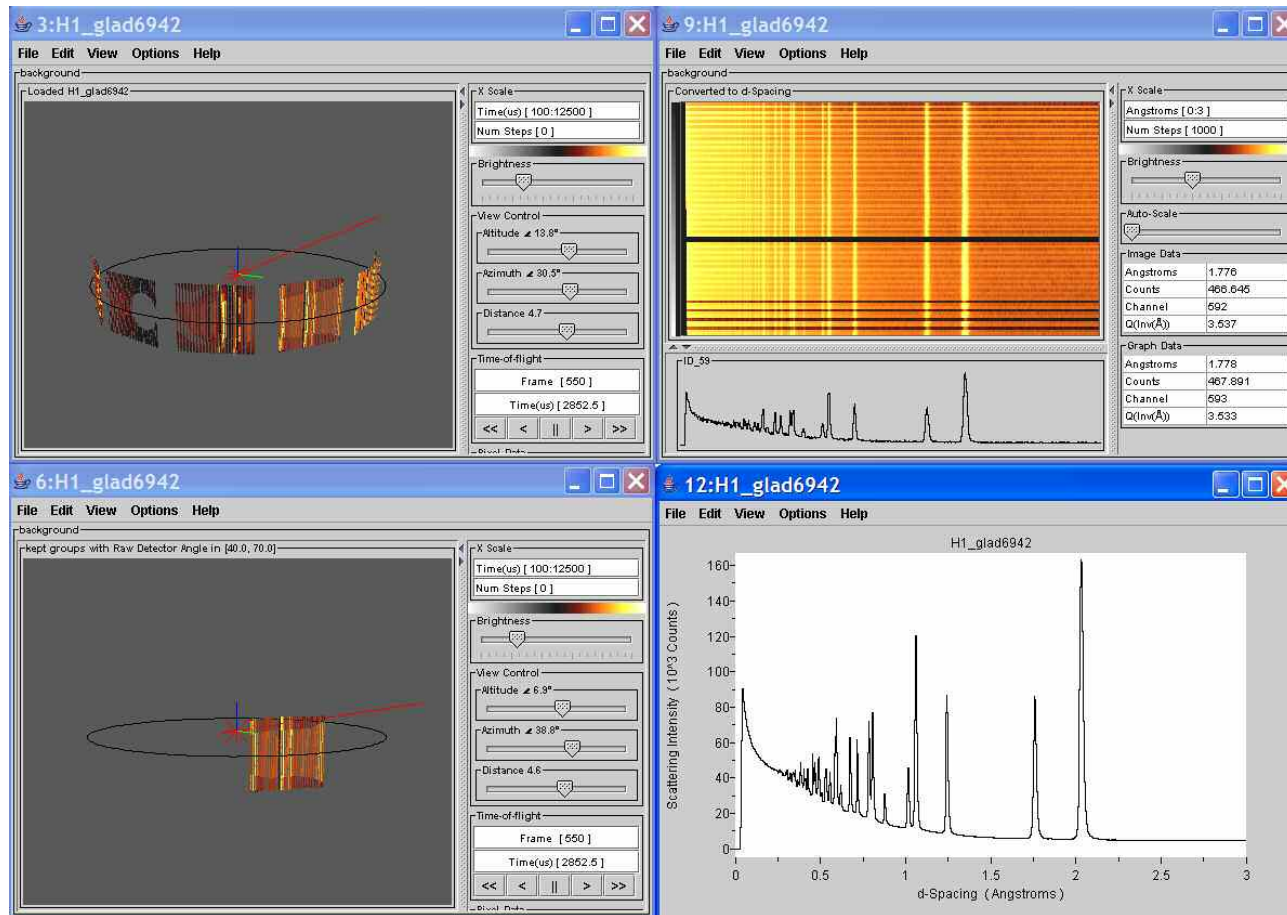
VTK

- Software system for 3D computer graphics, image processing, and visualization
- Used by thousands of researchers and developers around the world
- Written in C++
- Has Java wrappers
 - Also, Tcl/Tk, Python
- Free, Open Source



ISAW

- The primary tool for analyzing neutron scattering data at the IPNS
- Has an extensive and sophisticated interface



From: John Hammonds, IPNS

Java ?

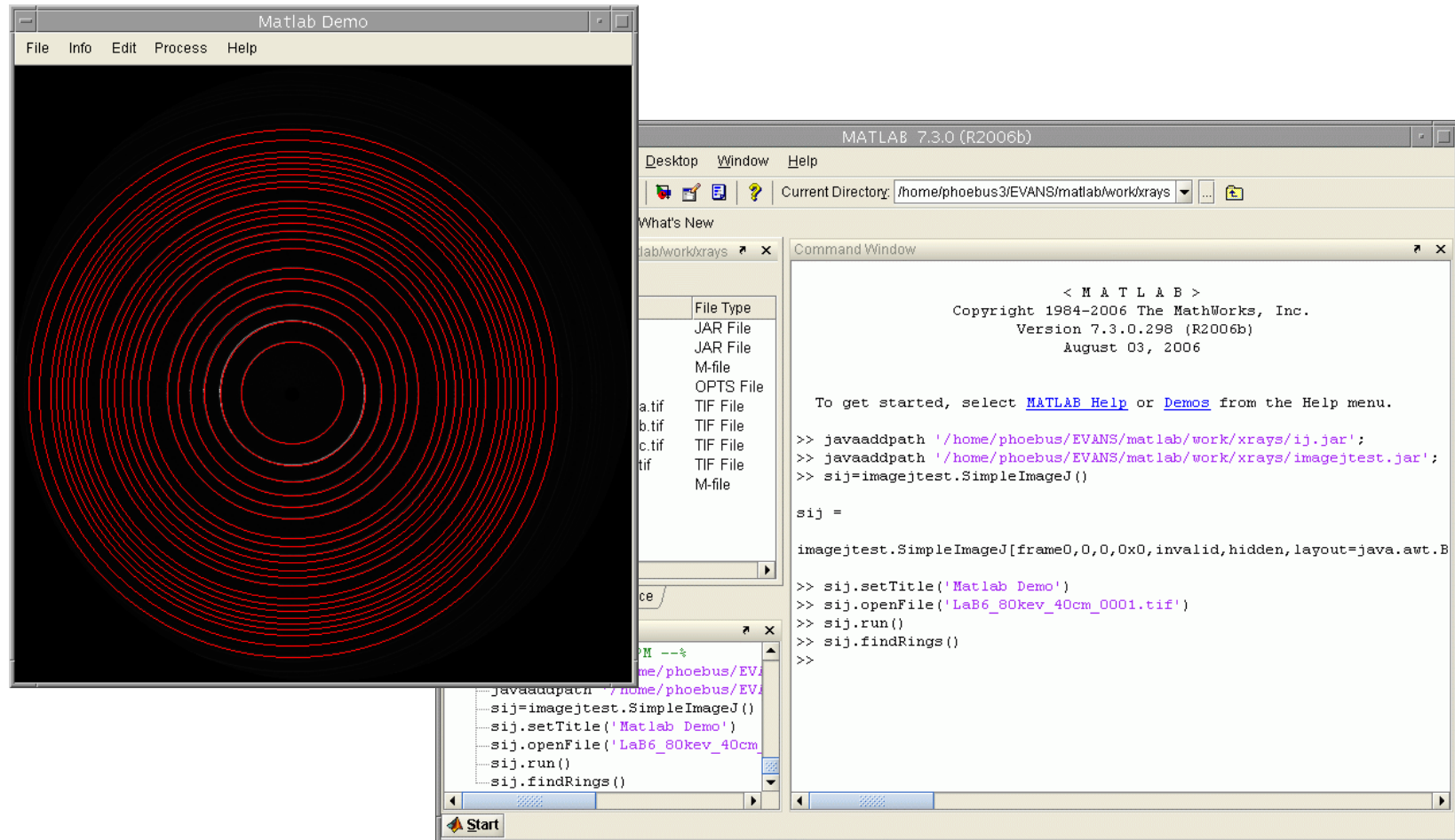
- Java has become a major language
- The reason is that most commercial development uses J2EE
 - There is money to be made improving Java and its tools
- Applications have performance approaching applications written in C
- There is already extensive scientific development in Java
- In my opinion, there is no other viable choice for high-quality, cross-platform, GUI development
 - Huge API
 - Write once, run anywhere
 - Easy to code (compared to C or C++, anyway)
 - Good performance
 - Excellent development tools

Java Development Tools

- Spell checks as you go
 - No “write – compile – load – run – figure out what happened” cycle
 - Probably the one most significant productivity enhancement
- Provides content assist
 - Probably the next most significant productivity enhancement
- Compiles as you write
 - Cycle is now “write – run”
- Massive refactoring
 - E.g. Change a variable name in all your files in all your projects
- Wizards and Tools to help at every stage
 - E.g. Generate getters and setters for all your properties
 - E.g. Add and/or clean up imports
- The above are just a small sample
 - Some of these are available for other languages
 - But usually not at the level they are for Java

Java in Matlab

- Matlab has extensive support for Java
 - Your favorite software framework can also be used in Matlab

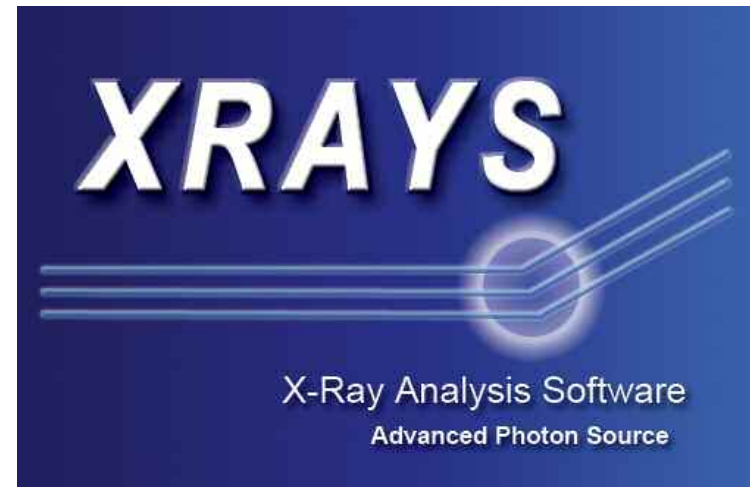


X-Ray Software Development at the APS

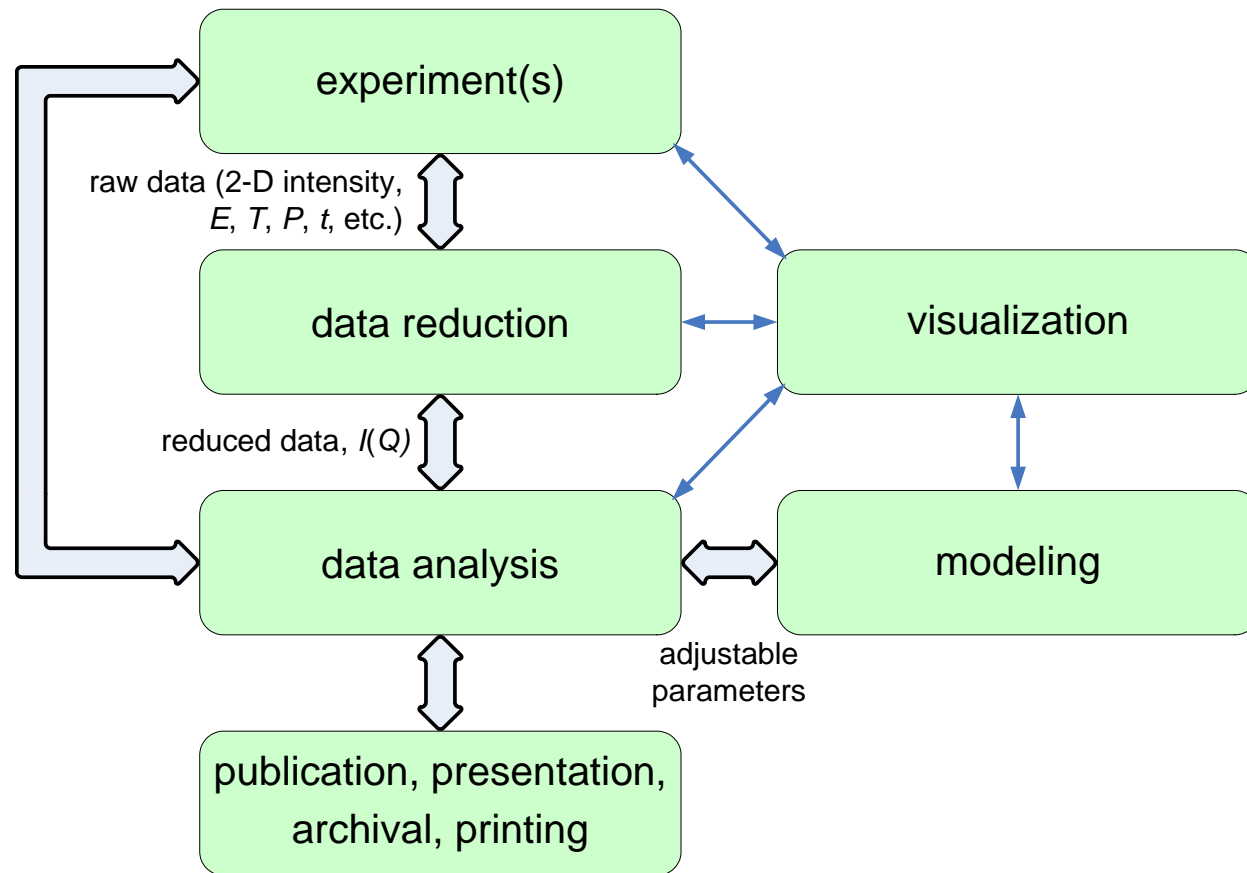
- Best described as “Uncoordinated”
 - Wide variety of languages
 - FORTRAN, C, C++, Perl, Tcl/Tk, Python, Java, ...
 - Visualization relies on (different) commercial products
 - IDL, IGOR, Matlab, ...
 - Each beamline tends to do its own thing
 - Modeling and Analysis is not well integrated with Data Acquisition
 - Lack of real-time data reduction
 - Little high-performance computing
 - Little remote access
 - No common data format
-
- A Scientific Software Section was formed to help remedy this situation

XRAYS

- Stands for X-Ray Analysis Software
 - (or X-Ray Software)
- It is expected to grow into a large suite of analysis and visualization applications
- These will include:
 - Scientific workbench program
 - New analysis and visualization applications
 - Updating and coordination of existing analysis and visualization applications
 - A framework of software routines that developers can use to write applications
- It currently consists mostly of exploration and prototype applications
 - This is the groundwork for what we really want to do
 - More than 1200 Java source files in 60 projects
 - 38 Java projects intended for distribution (gov.anl.xrays.xxx)
 - 10 ready-to-deploy features (collections of projects) in 4 categories



We Want to Manage the Entire Experimental Data Flow



Eclipse

- Eclipse is an Open Source community
- It was started in 2001 by IBM
 - IBM donated a lot of research
 - Controlled the early development, but later relinquished control
- It is now controlled by the Eclipse Foundation
 - Strategic members contribute up to \$500K and 8 developers
 - Currently 17 strategic members
 - Currently more than 150 developers
- Out of the box it looks like a Java IDE (Integrated Development Environment)
- It is really a Plug-in manager
 - That happens to come with Java Development plug-ins.
 - You can make it be most anything you want

XRAYS Rationalization for Eclipse

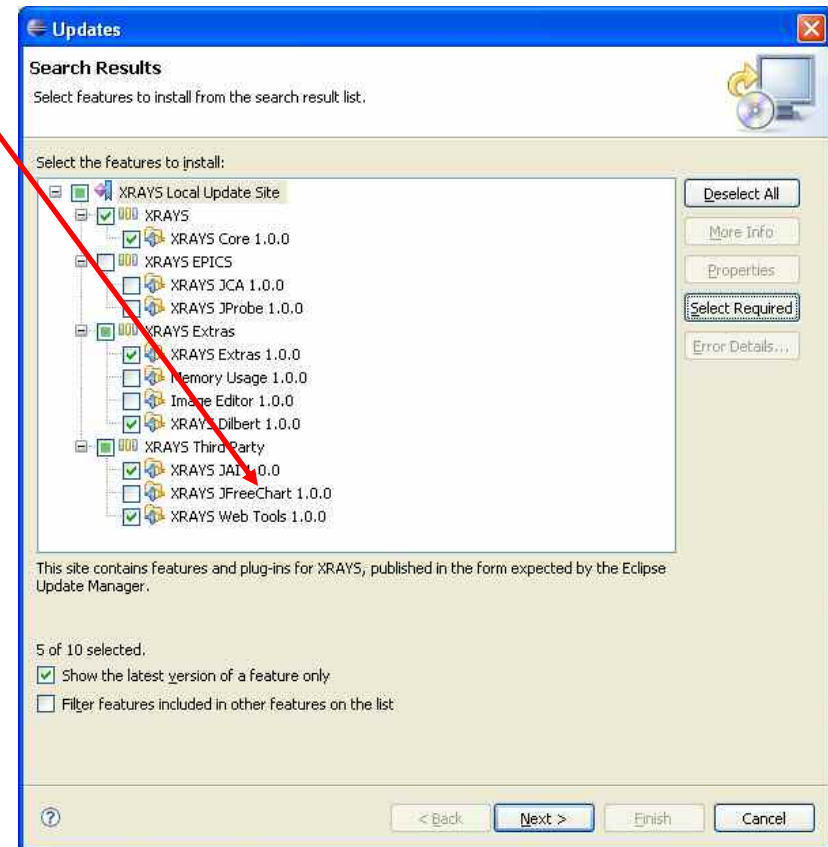
- Providing coordination is a primary goal
- Resources are limited
- Have to choose something
 - Eclipse seems like the best choice
 - Powerful, flexible, extensible
 - Open-source
 - Huge community with many projects
- Java development environment leads to high productivity
- Deployment via plug-ins appears to solve many problems

- We intend to use Eclipse, not as an IDE, but as a workbench
 - Something users will use

- Downsides
 - Most x-ray beamline staff and users are not using Eclipse now
 - 95% will be unhappy [with anything we do]

Deployment is a Major Reason for Using Eclipse

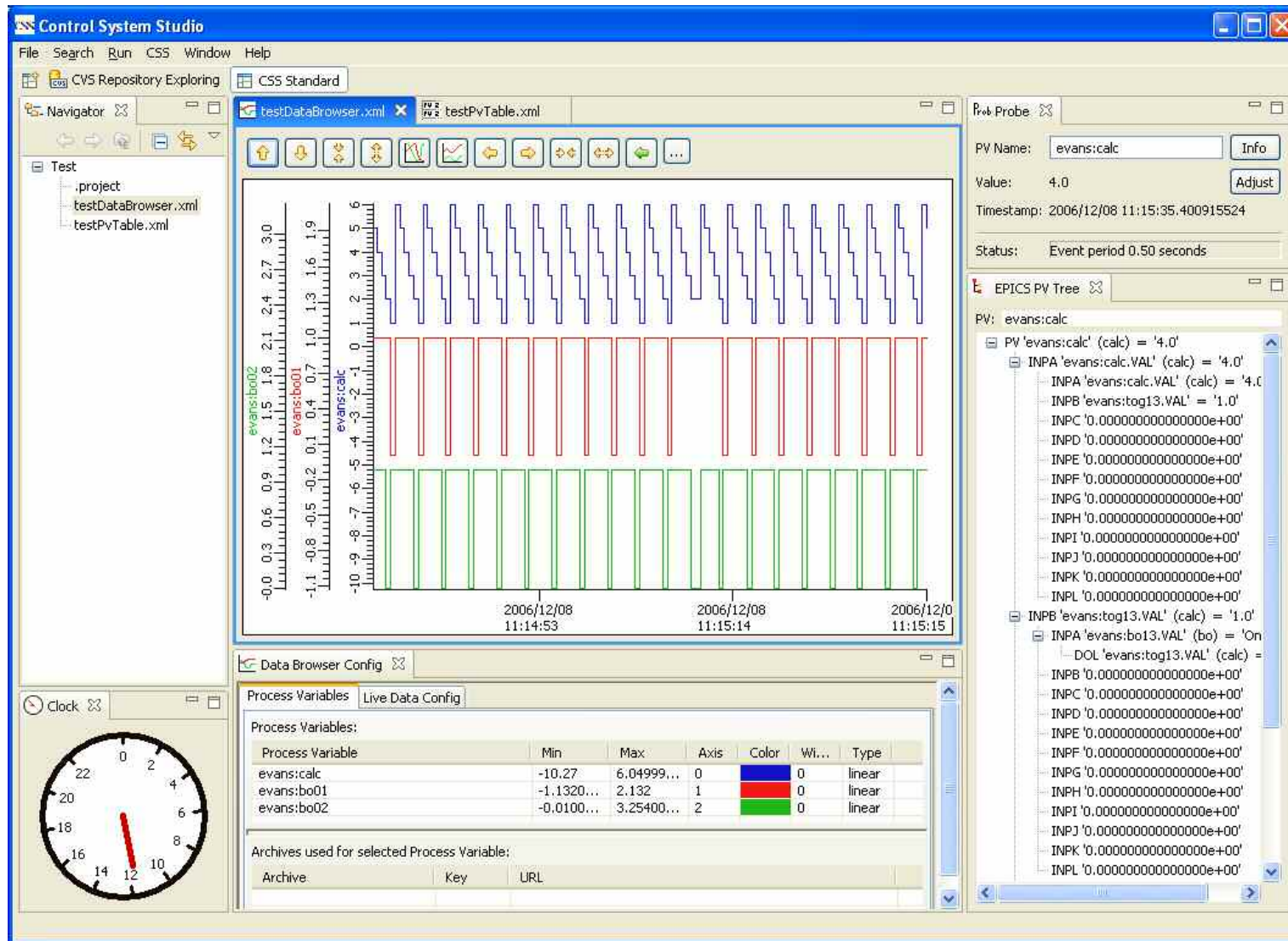
- Both Java and Eclipse are multi-platform
- Updates are easily made through the Eclipse update mechanism
- You can wrap 3rd party applications in your own plug-ins
 - For example:
The Feature “XRAYS JFreeChart” contains gov.anl.xrays.jfreechart which wraps JFreeChart
 - Including DLLs and Shared Objects
- Guarantees they are versions that work with your applications on all supported platforms
- Makes it easy for the user to install and update both your stuff and the 3rd party stuff



Eclipse for Users, not Developers

- We intend to use Eclipse as a workbench
- Something a user can come in and be up and running with in a short time
 - Probably with community help
- Each user can use and customize it in his or her own way
 - (That is what Eclipse provides)
- They will probably use it for more than one thing
 - That is why the layout by Perspective is important
 - You just switch perspectives to change tasks
- I think this paradigm is better than using RCP applications
 - You provide the plug-ins
 - The user manages his Workbench as he or she pleases

EPICS Control System Studio



EPICS IDE : IOC Development

The screenshot displays the Eclipse IDE interface for EPICS IOC development. The main editor shows the `st.cmd` file with the following content:

```
1#!/.../bin/cygwin-x86/test
2
3## You may have to change test to something else
4## everywhere it appears in this file
5
6##< envPaths
7
8## Register all support components
9dbLoadDatabase("../db/test.dbd",0,0)
10test_registerRecordDeviceDriver(pdbbase)
11
12## Load record instances
13dbLoadRecords("../db/dbExample1.db","user=evansHost")
14dbLoadRecords("../db/dbExample2.db","user=evansHost,no=1,scan=1")
15dbLoadRecords("../db/dbExample2.db","user=evansHost,no=2,scan=2")
16dbLoadRecords("../db/dbExample2.db","user=evansHost,no=3,scan=5")
```

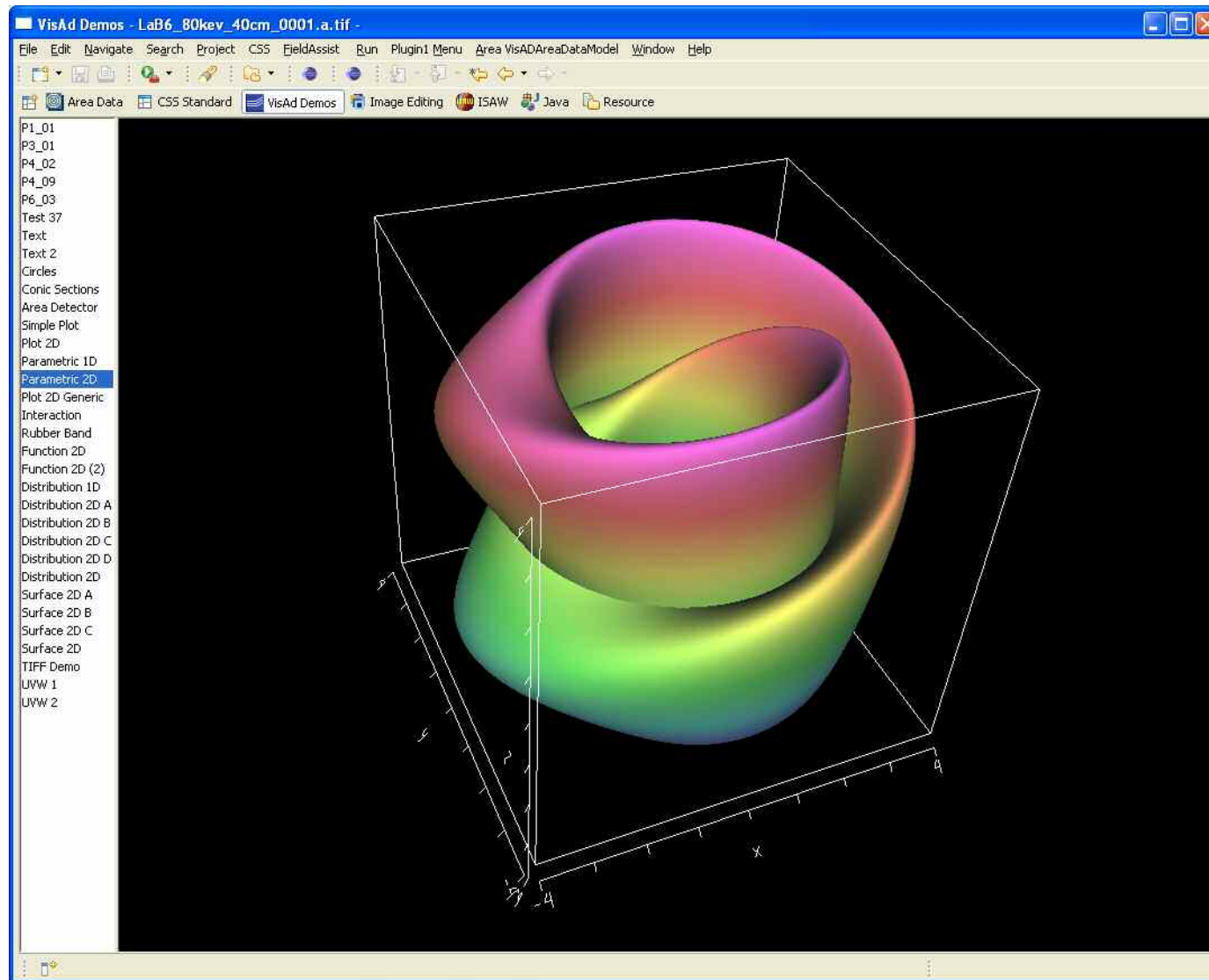
The left sidebar shows the project structure for `Test IOC`, including `bin`, `cygwin-x86`, `configure`, `db` (with `dbExample1.db`, `dbExample2.db`, `dbSubExample.db`), `dbd`, `include`, `iocBoot`, `iocTest` (with `Makefile`, `README`, `envPaths`, `st.cmd`), `iocTest2`, `lib`, `cygwin-x86`, `testApp`, `Db`, `src`, and `Makefile`.

The bottom console shows the execution output:

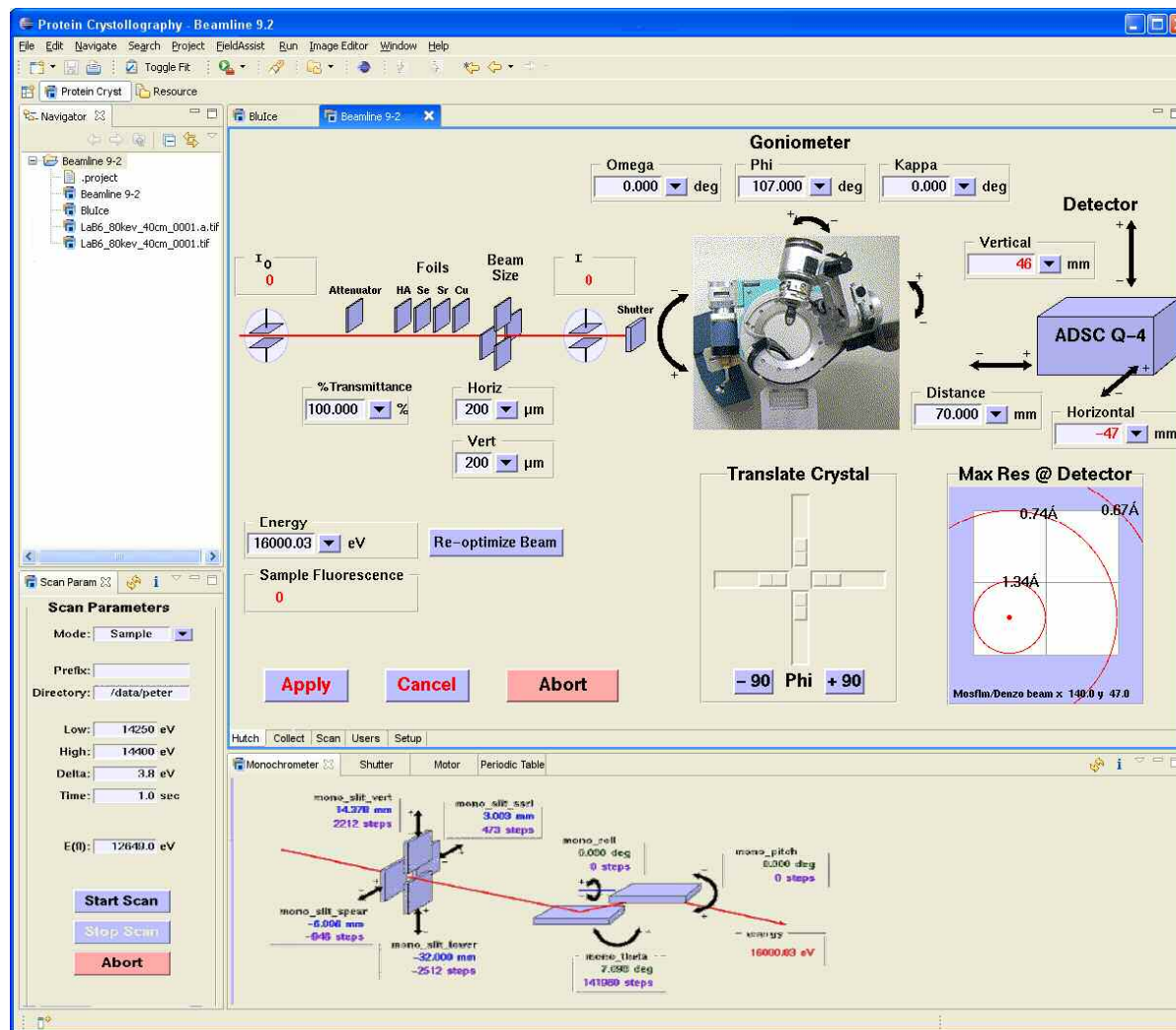
```
Test IOC [C/C++ Local Application] C:\Documents and Settings\evans\My Documents\Eclipse\Work\Test IOC\bin\cyg
dbLoadRecords("../db/dbSubExample.db","user=evansHost")
## Set this to see messages from mySub
#var mySubDebug 1
iocInit()
Starting iocInit
#####
### EPICS IOC CORE built on Apr 20 2006
### EPICS R3.14.8.2 $$Name: R3-14-8-2 $$Date: 2006/01/06 15:55:13
#####
iocInit: All initialization complete
## Start any sequence programs
#seq sncExample,"user=evans"
epics> db1
evansHost:aiExample
evansHost:aiExample1
evansHost:aiExample2
evansHost:aiExample3
```

The 'New Project' dialog is open, showing the 'EPICS Project' wizard. The 'Wizards' list includes `C++`, `CVS`, `Eclipse Modeling Framework`, `EMF Project`, `Empty EMF Project`, `EJB`, `EPICS` (selected), `Graphical Modeling Framework`, `J2EE`, `Java`, `Java Project`, and `Java Project from Existing Ant Buildfile`.

A Perspective Can be a Single Application



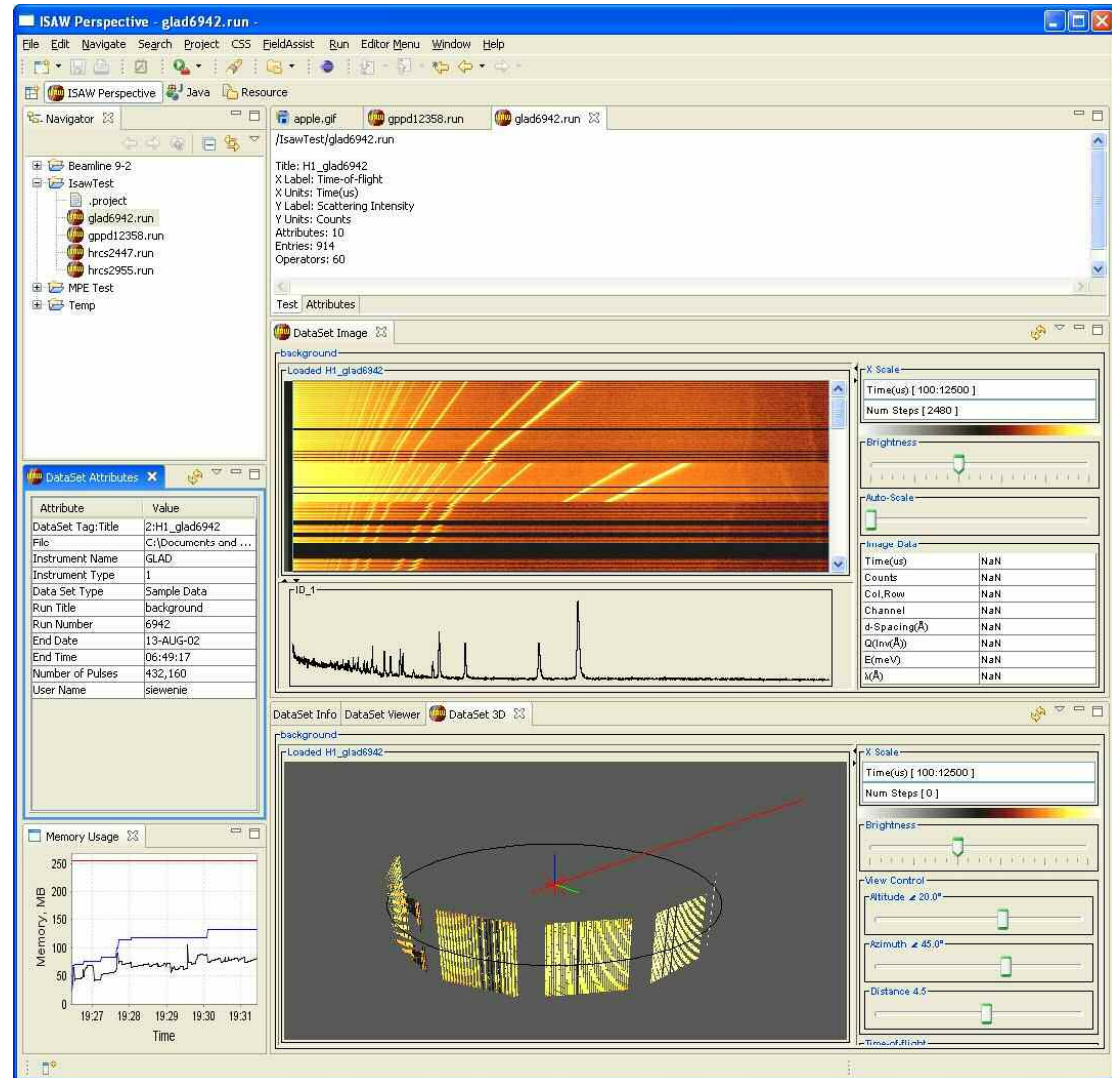
X-Ray Experiment



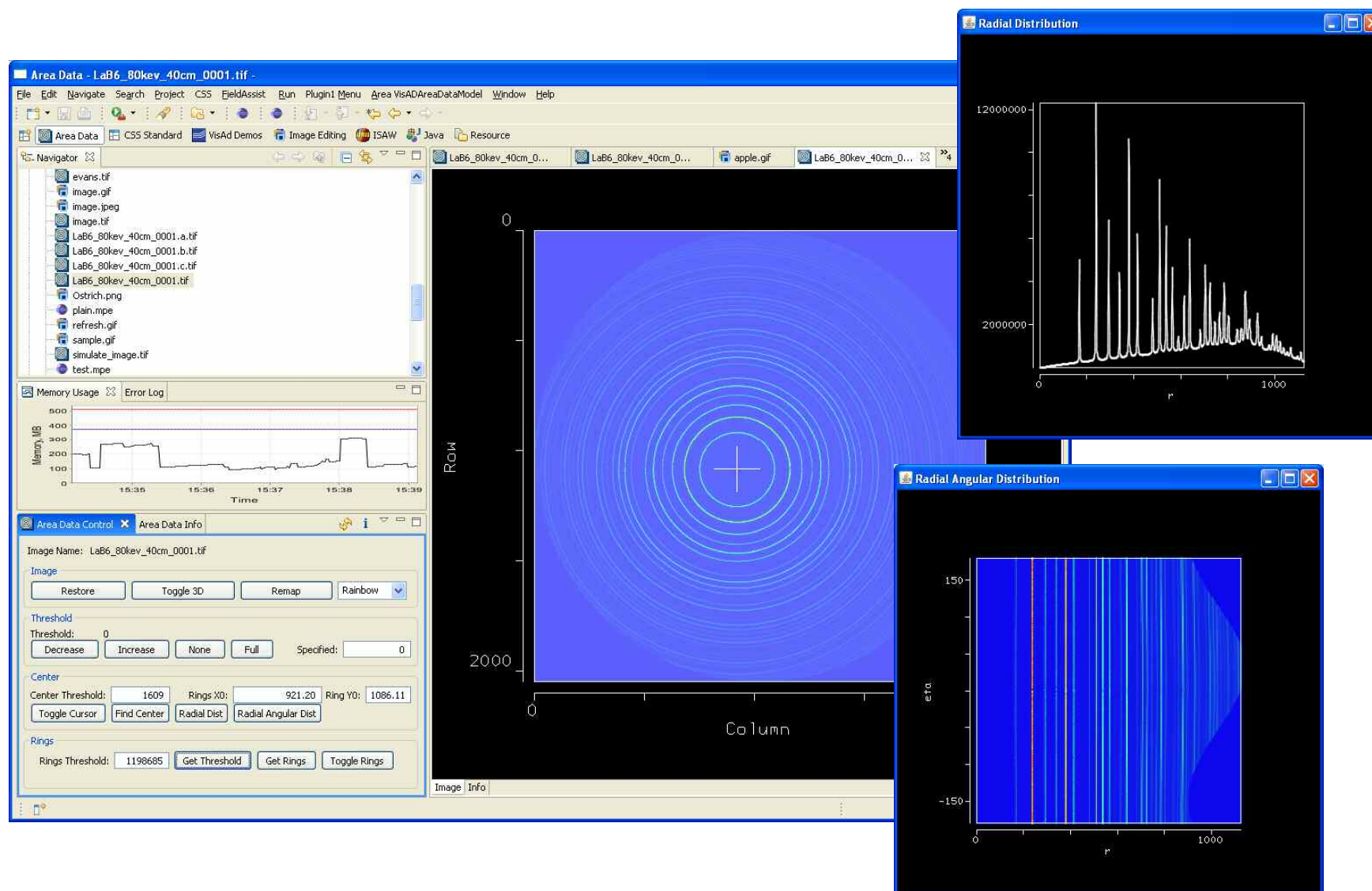
Images from: BLU-ICE and the Distributed Control System, NOBUGS III, January 2000

Prototype Implementation of ISAW

- Includes:
 - A Perspective
 - An Editor for ISAW DataSets
 - *.run*, *.isd*
 - Some Views
- All work together
 - Views change when the edited file changes



Area Data Editor - First Scientific Application



Thank You

*This has been a
Scientific Software Presentation*

Thank You

*This has been a
Scientific Software Presentation*